REMARKS

Claims 1-20 are pending in this application.

Claims 1-20 are rejected.

In the Office Action dated October 26, 2001, claims 1-20 are rejected under 35 USC §102(e) as being anticipated by Huras et al. The claims above have been amended for clarity, not in response to the '102 rejections. The claims are believed to be allowable over Huras et al. for the reasons that follow.

Amended claim1 recites a method in which a server handles a network connection. The method includes examining local server information to determine whether a client-to-server channel is still established; and aborting response preparation to a client request if the client-to-server channel is determined to be no longer established.

Huras et al. disclose a client process and a server process that are run on the same machine 100, such as a mainframe computer system (col. 4, lines 39-42). Huras et al. do not disclose a network environment. The system of Huras et al. uses shared memory and semaphores for communication between processes, not a network protocol such as TCP/IP. No network connections are formed between the client process and the server process. Huras et al. do not disclose a client-to-server channel or a server-to-client channel.

The system of Huras et al. is not faced with the latency problem that is addressed by the invention of claim 1. In the system of Huras et al., responses come quickly because the client and server processes are running on the same machine. Not so with the server that performs the method of claim 1. Because of

the increased probability of client disinterest, the server checks for possible client disinterest and avoids preparing an unnecessary response.

The system of Huras et al. does not abort response preparation, even if a client process is no longer interested in a response. Reference is made to Figures 1, 2A and 2B of Huras et al. Client process 150 writes request data to shared memory 250 (block 15), sets a valid request flag 252 to true (block 20), and posts a send semaphore 256 (block 25). This indicates to the OS that a process waiting on send semaphore 256 should be re-awakened. The OS places a server process 350 on a list of processes that are ready to run. There might be hundreds or even thousands of client processes (col. 1, line 19). There might also be hundreds or even thousands of associated server processes.

Now assume that as the next action the client process is terminated. The OS posts a send semaphore 256 (col. 8, line 59), but does not alter the value of the request flag 252, which remains as "true." The server process 350, when next allowed to run on the CPU, sees the request flag 252 as true indicating "valid request" (block 50), reads the request data in shared memory (block 60), processes the request (block 65), and writes the request data to shared memory 250 (block 70). Thus the server process responds to a client process, even though the client process is no longer interested in a response.

For these reasons, Huras et al do not teach or suggest the method of claim 1. Therefore, amended claim 1 should be allowable over Huras et al. In addition, dependent claims 2-7 should be allowable over Huras et al.

For the same reasons, independent claim 8 and its dependent claims 9-14, independent claim 15, and independent claim 16 and its dependent claims 17-20 should be allowable over Huras et al. Claim 21, which has been added to the

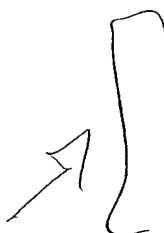application, should also be allowable over Huras et al.

Figure 6 of the drawings has been amended as indicated in red ink on the attached page. Amended Figure 6 is consistent with the specification. A replacement sheet will be submitted to the Official Draftsperson after amended Figure 6 has been accepted.

The specification has been amended. No new subject matter has been added.

An added claims fee has been incurred by the addition of claim 21. Authorization to charge the added claims fee is provided in the attached transmittal letter.

A petition for a ONE month extension of time is enclosed. The petition extends the period of response to February 26, 2002. Authorization to charge the petition fee is provided in the attached transmittal letter.

The Examiner is respectfully requested to withdraw the rejections and issue a notice of allowability. If there are remaining issues to be addressed, the Examiner is invited to contact the undersigned.

Respectfully submitted,

Hugh P. Gortler
Reg. No. 33,890

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on February 21, 2002.

Hugh P. Gortler

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400
(949) 454-0898

Date: February 21, 2002

<u>VERSION WITH MARKINGS TO SHOW CHANGES MADE</u>

1. In a server, a method of ~~responding to~~<u>handling</u> a ~~client request from a~~ network connection, the network connection including a client-to-server channel and a server-to-client channel, the method comprising:

examining local server information to determine whether the client-to-server channel is still established;<u> and</u>

~~inferring a state of the server-to-client channel according to whether the client-to-server channel is still established;~~

~~processing the client request if the inferred state indicates that the server-to-client channel is still established; and~~

<u>aborting response preparation to a</u> ~~terminating the~~ client request if the client-to-server <u>inferred state indicates that the server-to-client</u> channel is <u>determined to be</u> no longer established.

2. The method of claim 1, ~~wherein the client request is read from the client-to-server channel; and~~ wherein ~~the~~<u>a</u> state of the server-to-client channel is inferred <u>according to whether the client-to-server channel is still established; and wherein the response preparation is aborted if the server-to-client channel is inferred to be closed.</u> ~~after the client request has been read.~~

3. The method of claim ~~2~~<u>1</u>, wherein the server includes a read buffer; wherein the <u>client</u> request is read from the read buffer; and wherein the read buffer is then probed to determine whether the client-to-server channel is still established.

4. The method of claim 1, wherein the server maintains local information about the state of the ~~connection~~<u>client-to-server channel</u>; wherein a specific state

of the ~~connection~~ <u>client-to-server channel</u> is determined by examining the local information; ~~wherein the client request is read and processed if the local information indicates that the connection is not in the specific state;~~ and wherein the ~~request is not processed~~ <u>response preparation is aborted</u> if the local information indicates that the <u>client-to-server channel</u> ~~connection~~ is in the specific state.

5. The method of claim 4, wherein the <u>client-to-server</u> ~~server-to-client~~ channel is <u>determined</u> ~~inferred~~ to be no longer established if the local information indicates that the <u>client-to-server channel</u> ~~connection~~ is in a "CLOSE_WAIT" state.

6. The method of claim 1, wherein the state of <u>the client-to-server channel is determined</u> ~~the server-to-client channel is inferred~~ by polling the local information <u>,</u> ~~, the local information being polled~~ while a response to the client request is being prepared, whereby ~~a request can be terminated~~ <u>response preparation can be aborted</u> while ~~the~~ <u>a request</u> ~~response~~ is being prepared.

7. The method of claim 1, further comprising generating an interrupt when <u>the client-to-server</u> ~~the server-to-client~~ channel is <u>determined</u> ~~inferred~~ to be no longer established, wherein a response to the client request is processed until the interrupt is generated.

8. A network server comprising:
a processing unit;
a network interface card; and
computer memory <u>programmed to cause the processing unit to</u> ~~encoded with an operating system including a routine for commanding the processing unit~~

to maintain a queue of connections based on connection requests received by the network interface card;

the computer memory being further encoded with a server program including a routine for commanding the processing unit to accept connections from the queue; examine local server information to determine whether a client-to-server channel of a given connection from the queue is still established; process a client request associated with the given connection if the client-to-to-server channel of the given connection is still established; and abort response preparation terminate the associated client request if the client-to-server channel of the given connection is determined to be no longer established.

9. The server of claim 8, wherein the client request is read from its associated client-to-server channel; and wherein a state of a server-to-client channel of the given connection is inferred according to whether the client-to-server channel is still established; and wherein the response preparation is aborted if the server-to-client channel is inferred to be closed after the client request has been read; the client request being processed or terminated according to the inferred state.

10. The server of claim 9, further comprising a read buffer; wherein the a client request is read into from the read buffer; and wherein the read buffer is then probed to determine whether the client-to-server channel is still established

11. The server of claim 8, wherein the memory includes local information about a state of the given connection client-to-server channel; wherein a state of the given connection client-to-server channel is determined by examining the local information; wherein the client request is read and processed if the local information indicates that the given connection is not in a specific state; and

wherein the ~~request is not processed~~response preparation is aborted if the local information indicates that the ~~given connection~~client-to-server channel is in the specific state.

12. The server of claim 11, wherein ~~a server-to-client~~the client-to-server channel ~~of the given connection~~ is determined ~~inferred~~ to be no longer established if the local information indicates that the client-to-server channel ~~given connection~~ is in a "CLOSE_WAIT" state.

13. The server of claim 8, wherein a state of the client-to-server ~~a server-to-client~~ channel is determined ~~of the given connection is inferred~~ by polling the local information_, ~~the local information being polled~~ while a response to the client request is being prepared~~, the client request being processed or terminated according to the inferred state, whereby a request can be terminated while the response is being prepared~~.

14. The server of claim 8, wherein the memory is ~~further~~ ~~encoded~~programmed with a routine for commanding the processing unit to generate an interrupt when ~~a server-to-client~~the client-to-server channel ~~of the given connection~~ is determined ~~inferred~~ to be no longer established, and wherein a response to ~~the~~ a client request is processed until the interrupt is generated.

15. A network server comprising:
a processing unit;
first means for maintaining a queue of connections based on connection requests, each network connection including a client-to-server channel and a server-to-client channel;
second means for accepting connections from the queue;

third means for examining local server information to determine whether the client-to-server channel of a given connection from the queue is still established; <u>and</u>

~~fourth means for processing a client request associated with the given connection if it is determined that the client-to-server channel of the given connection is still established; and~~

<u>fourth</u> ~~ifth~~ means for <u>aborting response preparation</u> ~~terminating the associated request~~ if it is determined that the client-to-server channel of the given connection is no longer established.

16. An article ~~of manufacture~~ for a network server including a processing unit and a network interface card, the article comprising:

computer memory; <u>and</u>

~~an operating system routine encoded in the computer memory, the operating system routine, upon execution, commanding the processing unit to maintain a queue of connections based on connection requests received by the network interface card; and~~

a server program encoded in the computer memory, the server program ~~including a routine for~~ commanding the processing unit to accept <u>network</u> connections ~~from the queue~~, each connection having a client-to-server channel and a server-to-client channel; examine local server information to determine whether the client-to-server channel of a given connection from the queue is still established; ~~process a client request associated with the given connection if the client-to-server channel of the given connection is still established; and~~ ~~terminate~~ <u>abort response preparation</u> ~~the associated request~~ if the client-to-server channel of the given connection is <u>determined to be</u> no longer established.

17. The article of claim 16, ~~wherein the client request is read from the client-to-server channel of the given connection; and~~ wherein a state of the server-

to-client channel of the given connection is inferred ~~according to whether the corresponding client-to-server channel is still established~~ after the client request has been read.

18. The article of claim 16, wherein the memory is further encoded with local information about a state of the given connection; wherein ~~a~~ the state of the given connection is determined by examining the local information; ~~wherein the client request is read and processed if the local information indicates that the given connection is not in a specific state;~~ and wherein ~~the request is not processed~~ response preparation is aborted if the local information indicates that the client-to-server channel of the given connection is in the specific state.

19. The article of claim 16, wherein a state of the client-to-server channel of the given connection is determined by polling the local information, the local information being polled while a response to ~~the~~ a client request is being prepared, ~~whereby a request can be terminated while the response is being prepared~~.

20. The article of claim 16, wherein the memory is further encoded with a routine for commanding the processing unit to generate an interrupt when the client-to-server channel of the given connection is determined to be no longer established, and wherein a response to ~~the~~ a client request is processed until the interrupt is generated.